1. **Freeze top menu bar**: Apply position: fixed to the top menu in the Vue.js component.
2. **Change date fields**: Use Vue.js date picker components and adjust field types accordingly.
3. **Modify phone number registration field**: Update the input type and apply a mask for phone number formatting.
4. **Add validation to email fields**: Implement frontend email validation with regex and Vue.js form validation rules.
5. **Implement caution notifications**: Use Vue.js notifications or alerts to display caution messages.
6. **Create a notification window**: Build a modal or toast component for notifications using Vue.js.
7. **Save data in filled fields**: Implement v-model or Vuex to persist data across components.
8. **Add “Back” button to all pages**: Add router navigation for "Back" button functionality.
9. **Modify product catalogue pages**: Update Vue.js templates and components to reflect new design changes.
10. **Customer registration requirement**: Enforce registration checks in the Vue.js routing logic.
11. **Child selection and relation functionality**: Create Vue.js components for child selection and parent-child relationship forms.
12. **Develop two Home Page versions**: Implement i18n or conditional rendering for Russian and German home pages.

**Backend (Python):**

1. **Set password requirements**: Implement password strength validation in the registration.
2. **Email verification after registration**: Integrate email verification logic using Django email sending methods.
3. **Resend verification email option**: Add API endpoint to trigger email resend functionality.
4. **Modify course catalogue pages**: Update course-related database queries and templates.
5. **Trial lesson day selection**: Implement a field and validation for trial lesson date selection in the backend.
6. **Parent's info & child relation**: Add fields for parent's info and child relationship in user models and update endpoints.
7. **Ensure "Send/Submit" button is active**: Check and enforce form validation rules in the backend before enabling the button.
8. **Application confirmation**: Implement API logic to confirm application submissions.
9. **Ensure "Buy" button is active**: Validate cart and payment details before activating the "Buy" button.
10. **Payment options selection**: Add and integrate payment gateway options in the backend.
11. **Unlimited subscription option**: Implement logic for unlimited subscription in the subscription management module.
12. **Validate event/course participation**: Check participant eligibility and enrollment status before confirming participation.
13. **Manage basket blocks**: Implement logic to handle and update shopping cart items.
14. **Generate and display QR codes**: Use a QR code library to generate and render QR codes for transactions or user-specific actions.